EURASIP Journal on
Wireless Communications and Networking
a SpringerOpen Journal

RESEARCH Open Access

Energy-aware topology control for reliable data delivery in solar-powered WSNs

Dong Kun Noh[1] and Junbeom Hur[2*]

Abstract

Solar power is a valuable source of power for wireless sensor networks, but it periodically requires an appropriate energy management strategy. We introduce a scheme that constructs and maintains a fault-tolerant wireless sensor network topology that can make the best use of solar energy. This topology control scheme is based on a simple model of the availability of solar energy and matches the connectivity of each node with the energy left in its battery. Operating locally, our scheme constructs and maintains a *k*-connected backbone of energy-rich nodes that handles most of the network's traffic reliably, without depleting the reserves of energy-poor nodes. Simulation results demonstrate the effectiveness of our scheme.

Keywords: Solar energy; Fault tolerance; Topology control; Sensor network

1 Introduction

In a mains-powered wireless sensor network(WSN), energy is not a constraint on achieving a specific level of performance, so as to meet goals such as throughput and reliability. In a battery-based network, however, attempting to meet all goals in full can shorten the network lifetime; it may be better to sacrifice throughput and reliability, rather than to deplete the batteries of the sensor nodes. For this reason, designers of battery-based WSNs have focused on reducing energy consumption, so as to prolong the network lifetime.

Recently, however, environmental energy has emerged as a feasible supplement to battery power for wireless sensors, when manual recharging or replacement of batteries is not practical. In many situations, a ready source of environmental energy is the sun. Solar energy has a power density of about 15 mW/cm^3 [1], which compares very favorably with other renewable energy sources. Of course, this high-power density is only available for part of the day; nevertheless, its contribution allows the designer of a WSN more scope to consider positive measures to improve performance, rather than having to concentrate on energy consumption. In this paper, we look specifically at the contribution that a fault-tolerant topology can make to network performance.

Topology control involves coordinating nodes' decisions regarding their transmission ranges, in order to create a network with a desirable connectivity, while restraining energy consumption or increasing network capacity, or both [2]. The need for a network-wide perspective distinguishes topology control from other node-level techniques for saving energy or increasing network capacity. Although topology control is achieved through individual nodes choices of transmission power level, which determine their neighbors, but the result is a global property of the entire network.

Most research on topology control [3-5] has been predicated on the assumption that some level of network connectivity is the primary property to be achieved. Topology control protocols have then been formulated to achieve this connectivity with the lowest transmission power at each node. The resulting networks meet requirements in terms of capacity, energy consumption, and interference. However, they are more

*Correspondence: jbhur@cau.ac.kr
[2]School of Computer Science and Engineering, Chung-Ang University, 84 Heukseokro, Dongjak-gu, Seoul 156-756, Korea
Full list of author information is available at the end of the article

susceptible to node failure, because reducing nodes' transmission powers also reduces the number of possible routes between any pair of nodes.

Sensor nodes are notoriously unreliable, and therefore, fault tolerance is important in most WSNs. It is especially desirable that network connectivity should be preserved for as long as possible when some of sensor nodes fail or run out of power. Recent research on topology control [6-9] has therefore considered both energy efficiency and fault tolerance.

Increasing fault tolerance requires more transmission power. Thus, the sun would be an attractive power source for a fault-tolerant WSN if we could match energy requirement with its availability, but the workload of a WSN and the weather are both difficult to predict. Instead, we propose a simple solar energy model, which requires no explicit forecasting of either energy supply or demand, but nevertheless is able to help individual nodes to make the best use of solar energy in contributing to the performance of the network as a whole.

Based on this energy model, we introduce a distributed and localized algorithm, called SolarTC, that determines the transmission power at each node that will make the best contribution to the fault tolerance of the whole network, while taking account of the residual energy available in that node. SolarTC has the following properties:

- *Energy-adaptive operation*: A node running SolarTC usually operates in fault-tolerant mode, in which it tries to obtain more connectivity by increasing its transmission power. If a node has insufficient energy to operate in fault-tolerant mode, it switches to energy-saving mode, in which it tries to maintain the minimum transmission power required to maintain marginal network connectivity, while to minimize possible blackout time.
- *Supporting fault-tolerant and energy-aware routing*: The robust fault-tolerant nodes operate as a backbone network, which provides as many paths as possible to the sink node. This network can support either fault-tolerant routing or energy-aware routing.
- *Minimizing the disadvantage of a fault-tolerant topology*: Existing fault-tolerant topology control techniques which increase the transmission ranges of all nodes reduces the capacity of a network and increases MAC layer contention, whereas SolarTC only increases the transmission range of fault-tolerant nodes.
- *Fully localized algorithm*: A node running SolarTC only uses information about its neighbors which are one or two hops distant. Information about two-hop neighbors is obtained from one-hop neighbors.

The rest of this paper is organized as follows. In the next section, we review some existing schemes for topology control in WSNs and introduce solar-powered WSNs. In Section 3, we describe our energy model of a solar-powered sensor node, and in Section 4, we explain our energy-conserving topology control scheme for enhancing fault tolerance. Section 5 discusses the proposed topology control that affects the network layer. We then evaluate the performance of our algorithm in Section 6, and conclude the proposed work in Section 7.

2 Background

2.1 Effects of topology control in WSNs

A topology control protocol creates and maintains a list of the immediate neighbors of each node, in a network. Thus, it is related both to routing and to the MAC (medium access control) layer of the protocol stack, as shown in Figure 1. A topology control protocol is not a routing algorithm, but it can trigger a route update if it detects that a node's neighbor list has changed significantly. This allows a routing protocol to respond more quickly to topology changes and thus reduces the rate of packet loss. Conversely, the routing protocol can trigger the execution of the topology control protocol if the former detects a lot of broken routes in the network, since this strongly suggests that the actual network topology has changed appreciably since the last execution of topology control.

A topology control protocol is responsible for selecting the transmission range of each node, which determines

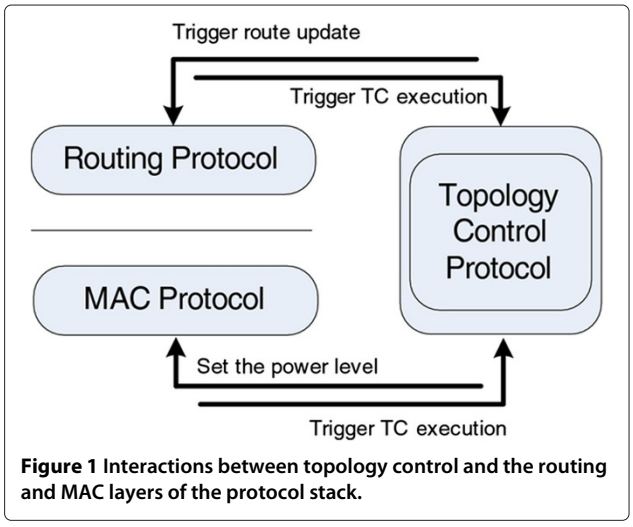

Figure 1 Interactions between topology control and the routing and MAC layers of the protocol stack.

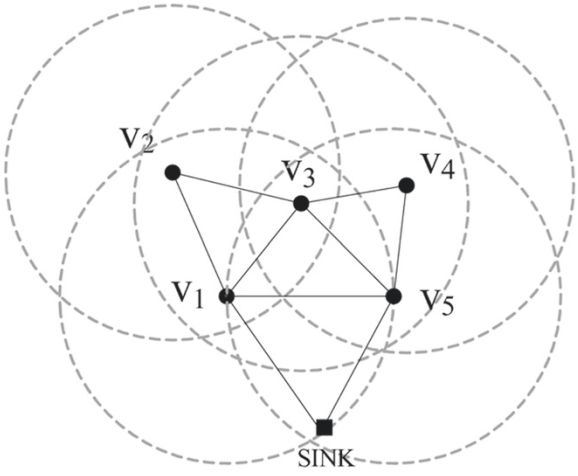

Figure 2 An example of a two-connected network.

the contention in the MAC layer. Like the routing protocol, the MAC protocol can trigger the execution of the topology control if it discovers new neighbor nodes by overhearing their traffic. This intimate relationship between topology control and the performance of the routing and the MAC layers makes topology control one of the most important issues in designing efficient WSNs.

2.2 Topology control for fault-tolerant WSNs

By reducing the transmission power of nodes to conserve the energy in nodes' batteries, most topology control protocols tend to reduce the numbers of routes between pairs of nodes. The reduced topology that results is naturally more susceptible to node failure.

It would be ideal to be able to construct a uniformly k-vertex connected (or k-connected) network, which would be k-1 fault tolerant, meaning that it is able to survive the failure of any k-1 nodes. Figure 2 shows an example of a two-connected network. Increasing the value of k requires a larger transmission range and sacrifices network lifetime and capacity. It is also likely to increase contention in the MAC layer. These disadvantages have to be traded against reliability.

A considerable amount of work [6-8] has been done on topology control protocols that try to create fault-tolerant networks, with the objective of minimizing power consumption while providing a specific level of connectivity. CBCT (fault-tolerant cone-based topology control) is a distributed and localized algorithm, proposed by Bahramgiri et al. [6], which achieves k-connectivity in a planar network by having each node increase its transmission power until either the maximum angle between any two neighbors is $2\pi/3k$, or it is transmitting at full power. Other authors [10,11] have tried to minimize the maximum transmission power used by any node. Ramanathan et al. [11] proposed a centralized greedy algorithm for achieving two-connectivity that iteratively merges pairs of mutually connected network components to construct the whole network. Li and Hou [10] introduced two variants of the local minimum spanning tree (LMST) algorithm [5] to address the k-connectivity problem: one of these (GLSS) is centralized and the other (FLSS) is distributed and localized. Both algorithms examine links in increasing order of length and only include the links needed to satisfy the k-connectivity condition. These procedures have been shown to be better than all other current schemes for reducing the peak power consumption of a node.

Most of the work that has been done on fault-tolerant topologies, including that described above, seeks to achieve uniform k-vertex connectivity between any two nodes in the network. This requirement is highly appropriate for *ad hoc* wireless networks, in which any two nodes can be the source and destination, because data is transmitted from the sensors to one or more sinks. Thus, it is not necessary to maintain a specific degree of fault tolerance between all pairs of sensors, but it is important to have fault-tolerant data paths between the sensors and the sink. Cardei et al. [12] addressed this problem with the k-ATC algorithm, which is tailored to heterogeneous battery-powered WSNs, in which data is forwarded from the sensors to so-called super-nodes through a k-connected network of fixed topology. Our algorithm also tries to achieve k-connectivity, but in the form of a k-connected backbone network with a structure that changes so that it always consists of energy-rich nodes.

2.3 Solar-powered WSNs

Corke et al. [13] articulated the principles involved in designing hardware for durable solar-powered wireless sensor networks. Minami et al. [14] designed a battery-less wireless sensor system for environmental monitoring, called *Solar-Biscuit*. Simjee and Chou [15] presented a solar-powered wireless sensor node, based on a supercapacitor, called *Everlast*. Jay et al. [16] described a systematic approach to build micro-solar power subsystems for wireless sensor nodes. However, most research on solar-powered WSNs, including the work mentioned above, has focused on node-level design topics such as hardware architecture and system analysis.

A few researchers [17-19] have been concerned with network-wide issues, such as latency and capacity, in solar-powered WSNs, but, to the best of our knowledge, we are the first to look specifically at topology control for enhancing fault tolerance in solar-powered WSNs.

3 Energy model for a solar-powered sensor node

An energy model for a solar-powered system requires both an expression of the expected harvest of solar energy, and an expression of the rate at which the system uses energy. The former is dependent on the season, the weather, and the location at which the system is deployed, and the latter depends on the rate at which data is sensed and transmitted, and the duty-cycle, which is the proportion of time for which a node is active. Unfortunately, these factors cannot be predicted precisely. We therefore propose a simple energy model [19] that is independent of these factors, and we will show how this model can be used to predict the amount of energy that a node can allocate to improving fault-tolerance.

Let P_{solar} be the average solar power acquired by a node, and let P_{sys} be its average rate of energy consumption. If the residual energy in the node is E_{residual}, then the expected time T_{full} before the node's battery will be full can be expressed as follows:

$$T_{\text{full}}(E_{\text{residual}}) = \frac{C - E_{\text{residual}}}{P_{\text{solar}} - P_{\text{sys}}}, \tag{1}$$

where C is the battery capacity.

Note that the battery will only charge when $P_{\text{solar}} > P_{\text{sys}}$. Otherwise, a node is not viable in the long term. We will explain how to control P_{sys} at the end of this section.

Even though the availability of solar energy varies from day to night and from one day to anther, a node should not blackout in the period before the battery is fully charged, as long as the residual charge in the battery at the start of this period satisfies the following condition:

$$E_{\text{residual}} \geq P_{\text{sys}} T_{\text{full}}(E_{\text{residual}}). \tag{2}$$

This is true even in the worst case, in which the solar energy arrives as late as possible. From Equations (1) and (2), we obtain $E_{\text{residual}} \geq \frac{P_{\text{sys}}}{P_{\text{solar}}} C$. This means that the system can run constantly in any environment if it has at least a threshold amount of energy $E_{\text{threshold}}$, where

$$E_{\text{threshold}} = \frac{P_{\text{sys}}}{P_{\text{solar}}} C. \tag{3}$$

If E_{residual} falls below $E_{\text{threshold}}$, then the node may have to shut down, and therefore, it needs to concentrate on saving energy. In SolarTC, a node in this situation operates as an energy-saving node, or ES-node. But when E_{residual} exceeds $E_{\text{threshold}}$, it starts to operate as a fault-tolerant node, or FT-node, which allocates some of its available energy to enhancing fault tolerance.

Determining $E_{\text{threshold}}$ requires a knowledge of P_{sys} and P_{solar}, which can both be estimated using moving averages. If $P_{\text{sys}}^{\text{new}}$ and $P_{\text{solar}}^{\text{new}}$ are the most recent samples of the rate of energy consumption and battery charging, then moving averages can be computed as follows:

$$P_{\text{sys}} = (1 - \theta_{\text{sys}}) P_{\text{sys}} + \theta P_{\text{sys}}^{\text{new}}, \tag{4}$$

$$P_{\text{solar}} = (1 - \theta_{\text{solar}}) P_{\text{solar}} + \theta P_{\text{solar}}^{\text{new}}, \tag{5}$$

where θ ($0 < \theta < 1$) controls the way in which the historical samples are considered: increasing θ reduces the contribution of older values.

The duty-cycle concept [20], which is often employed to allow a node to save energy, could easily be included in the calculation of P_{sys}. Since P_{sys} is proportional to the duty cycle, changing the duty cycle has a predicable effect on P_{sys}. This means that the designer of a WSN can control P_{sys} in an approximate way by adjusting the duty cycle. Therefore, $E_{\text{threshold}}$, which is dependant on P_{sys}, can also be controlled by varying the duty cycle, and this is important since the value of $E_{\text{threshold}}$ determines the number of FT nodes in the network.

4 Topology control

Definition 1 (*k*-FTN connectivity). *A graph G has k-FTN connectivity if, for any two FT-nodes n_1 and n_2, there are k pairwise vertex-disjoint paths from n_1 to n_2 on the backbone graph consisting of FT-nodes. Or equivalently, a graph is k-FTN connected if the backbone graph consisting of FT-nodes is still connected after the failure of up to $k-1$ FT-nodes.*

Basically, SolarTC$_k$ tries to construct a k-FTN backbone network consisting of FT-nodes. If that is not possible due to the low density of FT-nodes (we will discuss this in Section 4.3), it attempts to keep the connectivity of the backbone network as close to k as possible. We will explain the SolarTC$_k$ algorithm in more detail in this section, which uses the notation of Table 1.

4.1 SolarTC$_k$ algorithm

When sensor nodes are initially deployed, each node n_i starts by constructing its localized neighborhood $N_{\text{all}}(i)$ by exchanging 'hello' messages with all nodes that are within its maximum transmission range $r_i^{\max}$, and then it determines the minimum transmission power required to reach each neighbor. Next, node n_i runs the localized minimum spanning tree(LMST) algorithm [5], as shown in Figure 3, which has been shown to achieve connectivity for every feasible topology and has a very low overhead of n messages for a network of n nodes [5]. LMST initially creates a network with 1-connectivity, and we refer to the transmission power required by each node n_i to achieve

Table 1 Notation for SolarTC$_k(i)$ and FTTC$_k(i)$

Symbol	Meaning
$E_{\text{residual}}(i)$	Amount of residual energy at node n_i
$E_{\text{threshold}}(i)$	Energy threshold for changing mode, $(P_{\text{sys}}(i)/P_{\text{solar}}(i))C$
m_i	1 if sensor node n_i is in fault-tolerant mode, otherwise 0
s_i	1 if sensor node n_i has decided its operating power, otherwise 0
r_i, p_i	Current transmission range and power of sensor node n_i ($p_i = r_i^{\alpha}$), respectively
$r_i^{\max}, p_i^{\max}$	Maximal transmission range and power of node n_i, respectively
$N_{\text{all}}(i)$	$\{n_j \mid \text{dist}(n_i, n_j) \le r_i^{\max}\}$
$FTN_{\text{all}}(i)$	$\{n_j \in N_{\text{all}}(i) \mid n_j$ is in fault-tolerant mode$\}$
$FTN_{\text{covered}}(i)$	$\{n_j \in FTN_{\text{all}}(i) \mid \text{dist}(n_i, n_j) \le r_i\} \cup \{n_j \in FTN_{\text{all}}(i) \mid (r_i \le \text{dist}(n_i, n_j) \le r_i^{\max}) \wedge (n_i$ is k-connected to n_j in the localized view of the topology at $n_i)\}$
p_i^{LMST}	Transmission power of node n_i needed to maintain the LMST topology (1-connectivity)
$p_i^{\min}$	Transmission power of node n_i needed to reach the closest k fault-tolerant neighbors (FT-neighbors), or at least to maintain the LMST topology (1-connectivity)
$p_i^{\text{FTN}}(n)$	Transmission power of node n_i needed to reach the closest n FT-neighbors (including the sink), $p_i^{\text{FTN}}(0) = p_i^{\text{LMST}}$
Δp_i	Incremental power needed to reach the nearest node in $\text{FTN}_{\text{all}}(i)$-$\text{FTN}_{\text{covered}}(i)$

Algorithm 1 SolarTC$_k(i)$

```
1: if E_residual(i) > E_threshold(i) then
2:    m_i ← 1;
3: else
4:    m_i ← 0;
5: end if
6: if m_i = 1 then
7:    p_i ← p_i^max;
8:    broadcast_message (i, m_i);
9: end if
10: start timer t;
11: while t do
12:    if a broadcast message has been received then
13:       recalculate FTN_all(i);
14:    end if
15: end while
16: if m_i = 0 then
17:    p_i ← p_i^LMST;
18: else
19:    if | FTN_all(i) | < k then
20:       p_i ← max(p_i^LMST, P_i^FTN(|FTN_all(i)|));
21:    else
22:       invoke FTTC_k(i);
23:    end if
24: end if
25: return;
```

this connectivity as p_i^{LMST}. At this stage, node n_i has complete knowledge of its one-hop neighborhood which can be reached with p_i^{LMST}. However, this neighborhood may change if nodes are moved or go out of service, requiring LMST to be invoked regularly, or when certain events occurs, as shown in Figure 3.

After LMST has been run, node n_i invokes SolarTC$_k(i)$ periodically as shown in Figure 3. SolarTC$_k(i)$ begins by determining the mode m_i in which it will run from the amount of residual energy $E_{\text{residual}}(i)$ in the node's battery (lines 1 to 5). If $E_{\text{residual}}(i) > E_{\text{threshold}}(i)$, then m_i is set to 1 so that node n_i becomes an FT-node. It broadcasts this fact to its neighbors with its maximum transmission power (lines 6 to 9). There is then a back-off time, during which node n_i recalculates its list of FT-neighbors $\text{FTN}_{\text{all}}(i)$ from the messages received in response to its transmission (lines 11 to 15). $\text{FTN}_{\text{all}}(i)$ should be updated at every node, regardless of its mode m_i. If the node is an FT-node (m_i=1), then $\text{FTN}_{\text{all}}(i)$ will be used in the construction of the backbone network which consists of the FT-nodes. If the node is an ES-node, $\text{FTN}_{\text{all}}(i)$ can still be used by the routing scheme to find energy-rich neighbors (Algorithm 1).

After updating $\text{FTN}_{\text{all}}(i)$, node n_i determines its transmission power p_i from the value of m_i. If m_i is 0, then p_i is naturally the same as p_i^{LMST} (lines 16 to 17). Otherwise, the node must determine the value of p_i which corresponds to the closest approximation to preserve local k-FTN connectivity.

In order for a node to be k-connected, it must have at least k neighbors [21]. Similarly, for an FT-node to be k-FTN connected, it must have at least k FT-nodes within its transmission range. When $|\text{FTN}_{\text{all}}(i)| \ge k$, the routine FTTC$_k(i)$ is called to find the minimum transmission

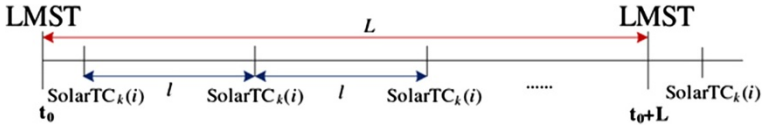

Figure 3 Periodic invocation of SolarTC$_k(i)$ to adapt to the changing energy status.

power p_i for node n_i which will create k disjoint-paths between n_i and each of its FT-neighbors n_j (line 22). Otherwise, p_i is set to $p_i^{\mathrm{FTN}}(|\mathrm{FTN_{all}(i)}|)$, so that n_i retains as many FT-nodes as possible in its neighbor list (line 20).

4.2 FTTC$_k$ algorithm

When the function FTTC$_k$(i) is run by node n_i, it computes $p_i^{\mathrm{FTN}}(k)$ and compares it with p_i^{LMST}. The larger of the two values then becomes that node's minimum transmission power $p_i^{\min}$, since $p_i^{\mathrm{FTN}}(k)$ is not necessarily sufficient to connect to all the neighbors of n_i in $N_{\mathrm{all}}(i)$. Node n_i then uses an iterative process to establish its actual transmission power p_i, starting from $p_i^{\min}$. This iteration ends when each node n_j in FTN$_{\mathrm{all}}$(i) is either within the transmission range r_i of node n_i or disjoint k-vertex paths are established between n_i and every n_j. The value of s_i is then set to 1 (Algorithm 2).

Algorithm 2 FTTC$_k$(i)

```
1: p_i^min ← max(p_i^LMST, P_i^FTN(k));
2: p_i ← p_i^min;
3: if p_i = p_i^max then
4:    s_i ← 1;
5: else
6:    s_i ← 0;
7: end if
8: recalculate (FTN_covered(i));
9: broadcast_message (i, m_i, p_i, s_i);
10: while s_i = 0 do
11:    calculate Δp_i;
12:    start timer t;
13:    while t do
14:       if a broadcast message has been received then
15:          recalculate (FTN_covered(i));
16:          recalculate (Δp_i);
17:          if FTN_covered(i) = FTN_all(i) then
18:             s_i ← 1;
19:             broadcast_message (i, m_i, p_i, s_i);
20:             return;
21:          end if
22:       end if
23:    end while
24:    p_i ← p_i + Δp_i;
25:    recalculate (FTN_covered(i));
26:    if FTN_covered(i) = FTN_all(i) then
27:       s_i ← 1;
28:    end if
29:    broadcast_message (i, m_i, p_i, s_i);
30: end while
31: return;
```

The algorithm terminates after at most $|\mathrm{FTN_{all}(i)}| - k$ rounds. In each round (line 10), both of FTN$_{\mathrm{covered}}$(i) and Δp_i are recalculated (lines 13 to 16) if a broadcast message has been received from an FT-neighbor in FTN$_{\mathrm{all}}$(i). This is straightforward since the message contains information about that neighbor and its one-hop FT-neighbors. The algorithm terminates when FTN$_{\mathrm{all}}$ (i) = FTN$_{\mathrm{covered}}$(i), and then node n_i broadcasts its p_i to all its neighbors (lines 17 to 20). If there is any FT-neighbor that is still out of range, the power level p_i is raised by the minimum increment Δp_i, which is sufficient to allow at least one more FT-neighbor in FTN$_{\mathrm{all}}$(i) - FTN$_{\mathrm{covered}}$ (i) to receive transmissions from node n_i (line 24). Then, the node recalculates FTN$_{\mathrm{covered}}$(i), since p_i has changed, and begins a new round, in which it seeks to discover whether the new value of p_i can reach all nodes in FTN$_{\mathrm{all}}$(i) (line 26). If it can, then the node broadcasts its new power level p_i (line 29) to all of its neighbors, and the algorithm terminates. Otherwise, the node broadcasts its current status (line 29) and starts another round.

4.3 Connectivity properties

Theorem 1. (k-FTConnectivity, N, of FTTC$_k$) *If the maximal power graph of all FT-nodes is k-connected, then the backbone network obtained by FTTC$_k$ is also k-connected.*

Proof. Let the set of all edges connecting FT-nodes constructed by FTTC$_k$ be E(FTTC$_{\mathrm{k}}$), and let the set of all edges connecting FT-nodes in the maximal power graph be E(MaxGraph). It is immediately apparent that E(FTTC$_{\mathrm{k}}$) is a subset of E(MaxGraph). Suppose there is an edge (n_i, n_j) which is in E(MaxGraph(i)) but not in E(FTTC$_{\mathrm{k}}$(i)). The FTTC$_k$(i) algorithm running in node n_i will only remove edge (n_i, n_j) from E(FTTC$_{\mathrm{k}}$(i)) if it is k-connected to node n_j through the nodes in its transmission range r_i. This means that k independent paths between node n_i and node n_j already exist. Thus the network is still k-connected after removing this edge (n_i, n_j) from E(MaxGraph(i)). Therefore, the edges removed by the FTTC$_k$(i) algorithm do not reduce the connectivity of the original graph which corresponds to the maximum transmission power. □

This theorem only allows FTTC$_k$ to guarantee k-FTN connectivity if the maximum-power graph of the FT-nodes is k-connected, and this becomes more likely as the number of FT-nodes increases.

As explained at the end of Section 3, $E_{\mathrm{threshold}}$ can be roughly determined from the duty cycle. If a node reduces $E_{\mathrm{threshold}}$ by decreasing its duty cycle, then that node is more likely to become an FT-node, since

$E_{\text{threshold}}$ is more likely to drop below E_{residual}. This relation allows an administrator indirect control over the number of FT-nodes in the network. However, reducing the duty cycle can also prevent a node from sensing data over an adequate period. Therefore, $E_{\text{threshold}}$ needs to be chosen to meet the requirements of a particular application; if fault tolerance is more important than the amount of data acquired, then a lower value of $E_{\text{threshold}}$ is more suitable and vice versa.

Theorem 2. (Bi-directional Connectivity of SolarTC$_k$) *The topology constructed by SolarTC$_k$ has only bi-directional links.*

Proof. The topology constructed by LMST has already been proven to be a bi-directional graph [5]. It remains to prove that the new links added by the SolarTC$_k$ algorithm are bi-directional. SolarTC$_k$ and FTTC$_k$, only try to add more links when they are running on an FT-node. And the only candidate links are those which connect that node to other FT-nodes. Therefore, no new link from an FT-node to an ES-node can be added, and all the existing links between FT-nodes and ES-nodes will have been created by LMST and therefore have bi-directional connectivity.

It now only remains to prove that the links between FT-nodes are bi-directional. Suppose that node n_i and node n_j are both FT-nodes and that there is a link between them. The existence of the link (n_i, n_j) means that these two nodes are not k-connected, since the FTTC$_k(i)$ routine would eliminate the link (n_i, n_j) if node n_i were already k-connected to node n_j. Similarly FTTC$_k(j)$ will retain the link (n_j, n_i). Consequently, the link between node n_i and node n_j is bi-directional. □

Bi-directional connectivity is a very important property for a wireless network, since it is essential for link-level acknowledgment, which is necessary for the reliable transmission of packets over unreliable media. Bi-directionality is also fundamental to floor acquisition mechanisms in the MAC layer, such as the RTS/CTS mechanism in IEEE 802.11. The disadvantages of unidirectional connectivity are well described by Marina and Das [22].

4.4 Design considerations

4.4.1 Frequency of invocation of LMST

Any method of topology control used in WSNs should be able to accommodate the addition, removal, and movement of nodes. The addition or removal of a node is sure to change the connectivity of a network, and moving a node is also likely to do so. To cope with these changes, our scheme periodically invokes LMST, as shown in Figure 3. The period between invocations (L in Figure 3) should be carefully determined based on the frequency of changes to the configuration of nodes, the pattern of node failure, and the speed at which node are likely to move. These factors depend on the application, the type of nodes in use, and the environmental conditions; thus, it is better to rely on historical data.

4.4.2 Frequency of invocation of SolarTC$_k$

The interval between runs of SolarTC$_k(i)$, labeled l in Figure 3, also needs to be chosen carefully: once SolarTC$_k(i)$ has determined the mode of a node, that node keeps operating in the same mode until the next run of SolarTC$_k(i)$. Suppose that node n_i is an FT-node ($m_i = 0$) at the start of a period of l and that $E_{\text{residual}}(i)$ drops below $E_{\text{threshold}}(i)$ before the end of that period, so that m_i becomes 0 and the node becomes an ES-node. Even so, node n_i should keep operating as an FT-node until the end of the period, so as to avoid damaging the existing topology before the next run of SolarTC$_k(i)$ constructs a new topology. Therefore, the value of l directly affects system performance.

Reducing the value of l helps SolarTC$_k(i)$ to reflect recent changes in the status of the nodes in a new topology, but it also incurs a very significant overhead. A larger value of l reduces this overhead but may prevent nodes from operating stably. For instance, an FT-node which needs the criteria to become an ES-node shortly after the start of a period may experience a blackout if it is forced to operate as an FT-node for too long. The optimal value of l depends on the characteristics of the nodes, the applications, and environmental conditions. The only feasible way of choosing l is by experiment.

4.4.3 Preventing repeated changes of mode

As explained in Section 4.1, the value of m_i for node n_i is determined from the relative values of $E_{\text{residual}}(i)$ and $E_{\text{threshold}}(i)$. However, frequent comparisons of $E_{\text{residual}}(i)$ and $E_{\text{threshold}}(i)$ may cause jitter in the value of m_i. Suppose that node n_i starts to operate as an FT-node as soon as $E_{\text{residual}}(i)$ exceeds $E_{\text{threshold}}(i)$. $E_{\text{residual}}(i)$ is very likely to sink below $E_{\text{threshold}}(i)$ almost at once. A similar but opposite effect is likely when a node becomes an ES-node. These repeated changes of mode degrade system reliability and performance. Therefore, SolarTC$_k$ uses an energy tolerance ψ to damp this oscillation. Lines 1 to 5 of SolarTC$_k$ then become

1: **if** $(m_i{=}0) \wedge (E_{\text{residual}}(i) > E_{\text{threshold}}(i) + \psi)$ **then**
2: $\quad m_i \leftarrow 1$;
3: **end if**
4: **if** $(m_i{=}1) \wedge (E_{\text{residual}}(i) < E_{\text{threshold}}(i) - \psi)$ **then**
5: $\quad m_i \leftarrow 0$;
6: **end if**

5 Effects of SolarTC$_k$ on the routing protocol

SolarTC$_k$ produces a backbone network consisting of FT-nodes. Theorem 1 has established that SolarTC$_k$ cannot guarantee k-FTN connectivity if the maximal power graph of FT-nodes is not k-connected. The density of the FT-nodes determines the topology of the backbone. It may be k-connected, simply connected, disconnected, or there may be no FT-nodes at all. Figure 4 shows examples of topologies constructed by the SolarTC$_2$ algorithm.

We will now provide some examples showing how SolarTC$_k$ can support fault tolerant and energy-aware routing. We use the energy-aware geographic routing (EGR) scheme [23], in which a node routes data to the most energy-rich of its neighbor nodes, in the direction of the sink node if possible. In order to use this scheme with SolarTC$_k$, we require each node to check its FT-neighbors before invoking EGR. The node then proceeds as follows:

- If there are FT-neighbors, the node applies EGR to those nodes alone.
- Otherwise, the node applies EGR to all its neighbors.

This modified routing scheme can be applied to each example in Figure 4. In Figure 4a, each node applies EGR to all its neighbors, since there are no FT-nodes. Data from node v_6, for example, might be routed to the sink along the path (v_6, v_5, v_4, v_3, v_2, v_1, sink), while data from v_9 might take the route (v_9, v_3, v_2, v_1, sink). These paths impose a heavy load on nodes v_1 and v_2, and are likely to bring about an energy imbalance. Moreover, if nodes v_1 or v_2 fail, the whole WSN becomes useless.

The network shown in Figure 4b has more energy, and SolarTC$_2$ makes nodes v_2, v_4, v_7, and v_{10} FT-nodes, and uses them to construct a backbone with 2-connectivity. In the modified scheme, node v_9 runs EGR and finds that nodes v_7 and v_{10} are its FT-neighbors. EGR will choose the most suitable of these FT-neighbors to receive data and probably selects node v_{10}, since it is closer to the sink node. In that case, data that has arrived at v_{10}, which is an FT-node, will be routed over the backbone to the sink, along the path (v_9, v_{10}, v_2, sink). Similarly, node v_6 can be expected to send its data long the path (v_6, v_7, v_4, sink).

In Figure 4c, nodes v_2, v_4, v_8, and v_{10} are FT-nodes. But SolarTC$_2$ cannot create a 2-connected network from these FT-nodes since they are not 2-connected, even using maximum power, because node v_4 cannot contact node v_8. So SolarTC$_2$ creates as many paths as possible among the FT-nodes. In this example, EGR running on node v_9 would send data to node v_{10} instead of node v_3, since node v_{10} is a FT-neighbor. Even though node v_{10} is not 2-connected to the sink, it has more energy than v_3, and its connectivity gives it more fault-tolerance. Node v_{10} routes the data

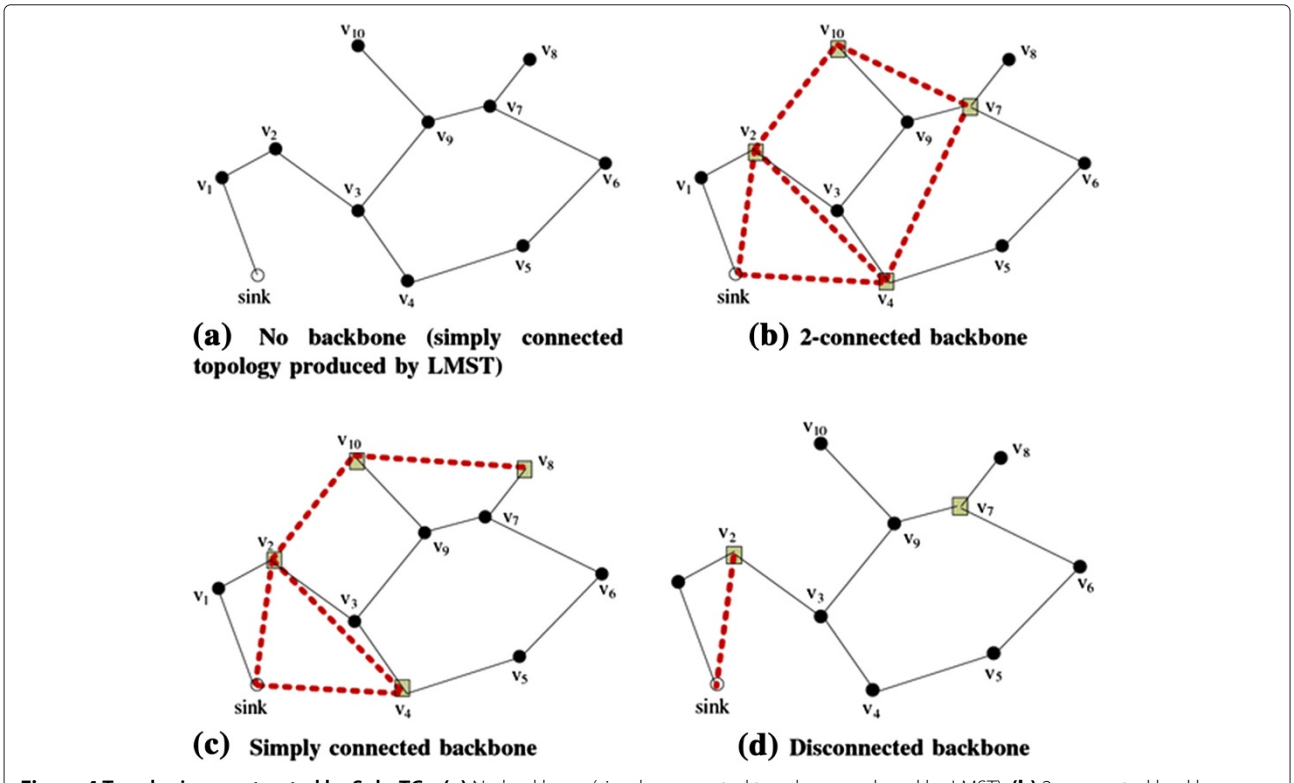

Figure 4 Topologies constructed by SolarTC$_2$. (a) No backbone (simply connected topology produced by LMST), **(b)** 2-connected backbone, **(c)** simply connected backbone, and **(d)** disconnected backbone.

to the sink over the backbone along the path (v_9, v_{10}, v_2, sink).

Lastly, there may be an isolated FT-node, like v_7 in Figure 4d. In this case, SolarTC$_2$ keeps the transmission power at v_7 as small as possible. This minimum power is $p_i^{\mathrm{FTN}}(0)$, which is the same as p_i^{LMST}. This does not affect the performance since v_7 cannot reach any other FT-nodes, even if it were to use its maximum transmission power $p_i^{\max}$. EGR will then make node v_6 route data to node v_7 instead of node v_5, since node v_7 is an FT-neighbor. But node v_7 has no FT-neighbors, and so it applies EGR to all its neighbors, and the resulting path is (v_7, v_9, v_3, v_2, sink). Although node v_7 is not connected to the backbone network and therefore cannot provide any fault-tolerance, it is beneficial for its neighbors to route data through node v_7 since it has enough energy to accommodate a high workload.

6 Performance evaluation

We have set up outdoor testbed for solar-powered sensor networks, as described in our previous publications [19,24]. Unfortunately, however, the Ethernet cards used in the testbed cannot support any change of transmission power which is an essential function for our scheme. Moreover, there are too few nodes in the testbed to evaluate the performance of fault tolerance. Inevitably, therefore, we designed an NS2 simulation for the performance verification.

6.1 Simulation

6.1.1 Energy model

We used solar-energy data obtained from our outdoor testbed [19,24] in Urbana, IL, USA, during the 15 days between 01 and 15 December 2009. The average energy harvested by a node during this period was 28.2Ah (at 12V). The energy consumption model that we constructed mimics the characteristics of our outdoor system [19] (including the energy consumed in sensing data, accessing memory, CPU operations, and so on) except for the energy used in data transmission. The amount of energy used for transmitting and receiving 1 bit of data over a distance d is expressed as follows [25]:

$$\begin{aligned} E &= E_{\mathrm{elec}}^{\mathrm{trans}} + \beta d^{\alpha} + E_{\mathrm{elec}}^{\mathrm{receive}} \\ &= 2E_{\mathrm{elec}} + \beta d^{\alpha}, \end{aligned} \tag{6}$$

where E_{elec} is the energy consumed by the electronics (J/bit); α is the path loss ($2 \le \alpha \le 5$); β is the energy used by the power amplifier in transmitting 1 bit over a distance of 1 m (J/bit/m); and d is the distance between the nodes (m). Table 2 shows the important parameters of this energy model, including the battery characteristics.

Table 2 Energy model parameters

Parameter	Value
Battery capacity	98 Ah
Battery output voltage	12 V
Energy harvested in 1 day	22.1 to 38.8 Ah (avg. 28.2 Ah)
α and β in Equation (6)	2 and 100

6.1.2 Traffic model

The simulation of SolarTC was tested on an application that runs on our outdoor testbed, so collect birdsong for studies of bird populations. We captured a pattern of traffic from the outdoor testbed to use in the simulation. We chose to explore a scenario in which nodes have different sensing rates, with ratios of 1:2:4 between them. This reflects real applications in which more interesting events usually occur at some locations than others. For example, in our application, sensors located near nests obtain more data than others. Each node in the simulation was randomly assigned one of the three sensing rates and creates traffic at that rate, unless that node is in sleep mode.

6.1.3 Simulated WSN

We simulated a WSN containing between 50 and 500 solar-powered nodes, spread randomly over an area of 800×800 m^2. The maximum radio range was set to 200 m, and the period between runs of SolarTC$_k$ was half an hour. We use the modified EGR scheme [23] described in Section 5. Tables 3 and 4 summarize the experimental and performance parameters of our simulation respectively.

6.2 Simulation results

Figure 5 shows how the residual energy E_{residual} and the energy threshold $E_{\mathrm{threshold}}$ vary under SolarTC$_4$ running on node n_{30}. Node n_{30} starts with an nearly half-charged

Table 3 Experimental parameters used in the simulation

Parameter	Value
$r_i^{\max}$	200 m
θ_{sys} and θ_{solar} in Equation (4)	Both 0.5
Period (I) of SolarTC$_k$ in Figure 3	30 min
Period (L) of LMST in Figure 3	1 day
Energy window ψ in Section 4.4.3	5 Ah
Level of fault-tolerance (k)	1, 2, and 4
Routing protocol	Modified EGR scheme
MAC protocol	802.11
Terrain	800 m × 800 m
Number of nodes	50 to 500 by 50
Number of sink	1
Node placement policy	Random

Table 4 Performance parameters of the simulation

Performance	Performance parameter
Degree of fault-tolerance	Average number of FT-nodes for different value of k
	Probability of k-FTN connectivity for different value of k
Energy efficiency	Average transmission range of the nodes
	Average proportion of blackout time
	Average throughput of the network

battery. For the first 4 days, the residual energy drops below the threshold and stays there due to a small rate of energy harvest. Thus, node n_{30} remains in energy-saving mode and SolarTC$_4$ retains the LMST topology without attempting to enhance fault tolerance. In each of the following days, there are always a few hours when E_{residual} is larger than $E_{\text{threshold}}$, and thus n_{30} operates as a fault-tolerant mode during these periods. Since energy is consumed more slowly when a node is operating as an ES-node, the changes in residual energy over the first 4 days exhibit the pattern labeled Type B in Figure 5, and over the next 5 days the changes follow Type C. Subsequently, the Type A pattern occurs when the node is always operating in fault-tolerant mode, and Type B when it is in energy-saving mode. We call this composite pattern Type C.

Figure 6 shows the average number of FT-nodes when the network is running SolarTC$_2$ and SolarTC$_4$, respectively. The larger value of k naturally incurs faster energy consumption by the FT-nodes. This quickly leads E_{residual} to fall below $E_{\text{threshold}}$, so most nodes returns to energy-saving mode immediately. Therefore, the average number of FT-nodes decreases as the value of k gets larger. Moreover, the high rate of energy consumption at each node increases P_{sys}, which raises $E_{\text{threshold}}$. This makes it less likely that a node will operate as an FT-node. As shown in Figure 6, between 35% and 50% of all nodes are FT-nodes when $k = 2$, but only 25% to 35% are FT-nodes when $k = 4$.

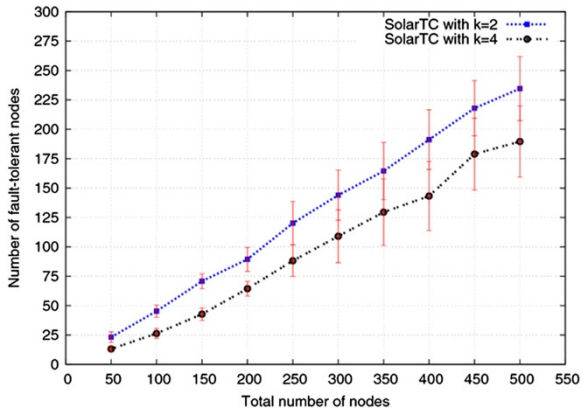

Figure 6 Variation in the average number of FT-nodes with k.

Figure 7 shows how the probability of achieving k-FTN connectivity varies with the number of nodes, for values of k of 2 and 4. We can immediately see that k-FTN connectivity increases with the number of nodes, regardless of the value of k. This is because the number of FT-nodes is proportional to the total number of nodes, as shown in Figure 6. It is also apparent that larger values of k require more FT-nodes to preserve k-FTN connectivity, but larger values of k mean that there are actually fewer FT-nodes, as shown in Figure 6. As a result, the probability of achieving k-FTN connectivity drops as the k increases. As shown in Figure 7, 2-FTN connectivity can almost be achieved when the network has over 150 nodes, and 4-FTN connectivity is possible with over 350 nodes.

These results are significant for the deployment of sensors in the field. For example, suppose that there are 150

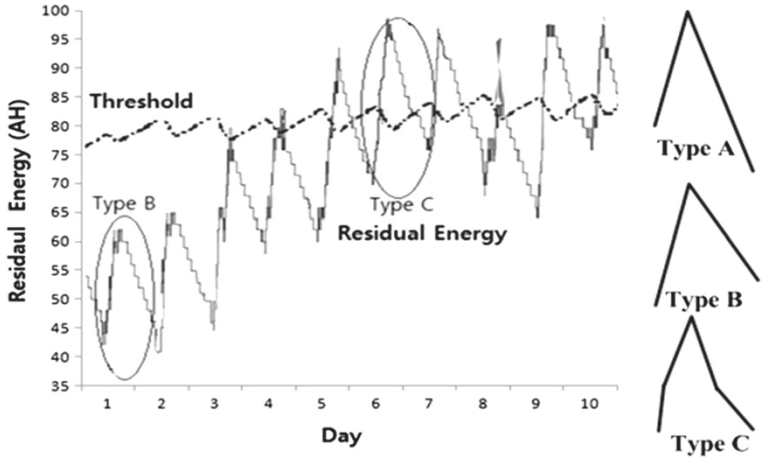

Figure 5 Simulated performance of SolarTC$_4$ in a sample node.

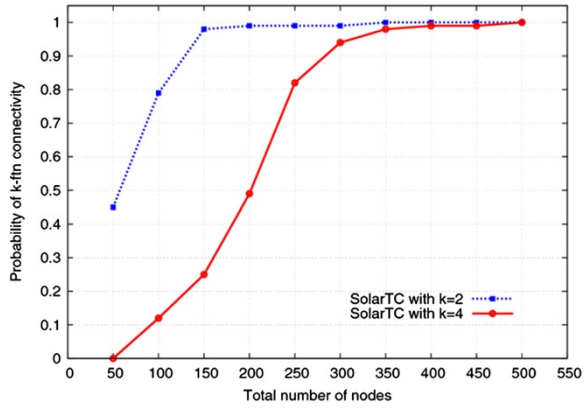

Figure 7 Probability of *k*-FTN connectivity for different values of *k*.

nodes and the value of k is set to 4. Then, each FT-node uses a high transmission power in an attempt to achieve 4-connectivity, as we see in Figure 8, but the probability of achieving 4-FTN connectivity is only about 20%. Thus this endeavor is likely to waste energy, and an attempt at k-connectivity should only be made if reliable transmission is at a premium.

FLSS [10] is an efficient algorithm for establishing an *ad hoc* network with k-connectivity. Since SolarTC only provides k-FTN connectivity, FLSS constructs a more robust network than SolarTC, for the same value of k. Therefore, it may not be fair simply to compare FLSS and SolarTC. But there is no topology control scheme more like SolarTC, as far as we know, so we compare the performance of FLSS with that of SolarTC, as a marginal reference.

Figure 9 shows the average transmission range of all nodes, running FLSS and SolarTC, with k set to 2 and 4. Both schemes require larger transmission ranges to achieve the higher value of k. But SolarTC requires a shorter range, and hence less power, than FLSS for the same value of k. But the difference is barely significant and may be caused by the schemes' different ways of using the energy available: FLSS simply tries to achieve k-connectivity and does not consider energy, whereas only the FT-nodes running SolarTC use residual energy to increase their transmission range, and SolarTC tries to achieve k-connectivity among those nodes alone. Therefore, as shown in Figure 8, the average transmission range of FT-nodes is larger than that of ES-nodes, and this difference increases when $k = 4$ (Figure 8b). Thus, the transmission range of ES-nodes used by SolarTC is much smaller than the range of an average node running FLSS. It is this asymmetric assignment of transmission range that gives SolarTC its edge.

We can make one more interesting observation from Figure 8b. Generally, the transmission range of each node decreases when there are more nodes, since the density of nodes is higher. In Figure 8b, however, the FT-nodes show an opposite trend, until the number of nodes reaches 250. We suggest that this is related to the probability of 4-FTN connectivity, shown in Figure 7. Since the network is unlikely to achieve 4-FTN connectivity with few nodes, SolarTC$_4$ increases the range of each node to get as close to 4-FTN connectivity as possible.

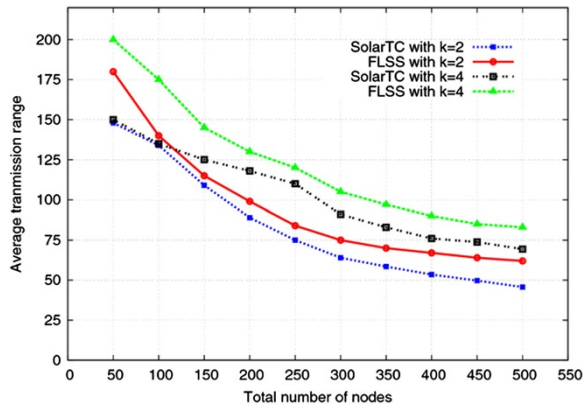

Figure 8 Average transmission ranges of FT- and ES-nodes.

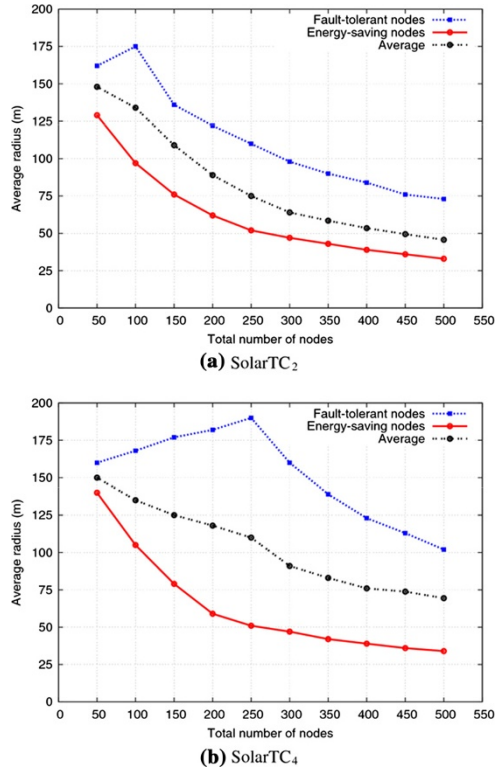

Figure 9 Average transmission range of all nodes, running FLSS and SolarTC. (a) SolarTC$_2$ and **(b)** SolarTC$_4$.

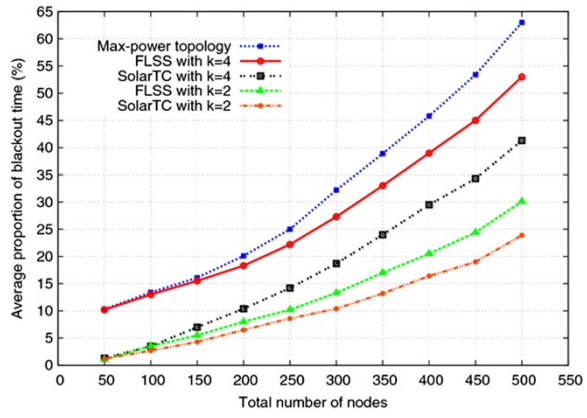

Figure 10 Average proportion of blackout time.

The proportion of blackout time at an average node for each scheme is shown in Figure 10. As the number of nodes grows, the average transmission range of each node decreases, as shown in Figure 9, but nodes' energy consumption increases since more data has to be relayed over more hop. Thus, the blackout time increases, as shown in Figure 10. Because SolarTC achieves a better balance than FLSS between workload and available energy across the network, there are fewer blackouts. Blackouts reduce a network's throughput and its capacity to acquire data, with a knock-on effect on the topology.

Finally, in Figure 11, we compare the throughput of the network at the sink node. It can be observed that increasing the fault-tolerance of the network by using a higher k reduces its throughput. This demonstrates the trade-off between the robustness of the network, manifest as the number of redundant routes, and its capacity. Additionally, SolarTC achieves a better throughput than FLSS, since it has less blackout time (because of greater energy efficiency) and fewer nodes with a relatively high transmission range (due to the increased probability of spatial reuse and reduced MAC layer contention).

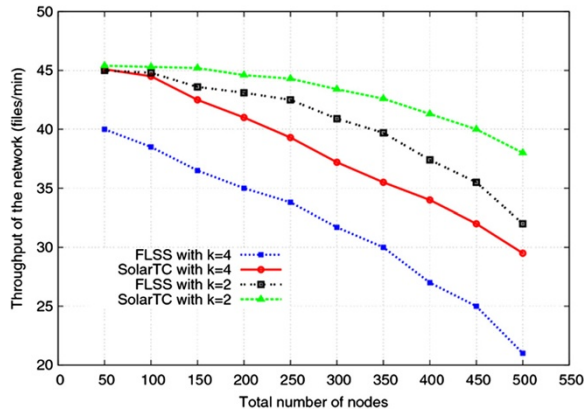

Figure 11 Throughput of the network.

7 Conclusions

It is not easy to make good use of solar energy in a WSN, because its availability depends on the time, season, and weather. We propose a simple solar energy model that requires no forecasting but nevertheless helps individual nodes to make the best use of energy and hence network performance improves, in particular fault-tolerance.

Based on this energy model, we designed a localized scheme for adaptive topology control which maintains k-connectivity from all the energy-rich nodes in the network to a sink whenever possible; otherwise, it tries to keep as many paths as possible between these energy-rich nodes, so as to support energy-efficient routing. This scheme also increase network stability by reducing the unscheduled blackout time of solar-powered sensor nodes.

Competing interests
The authors declare that they have no competing interests.

Acknowledgements
This research was supported partly by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (2011-0012996), partly by Industrial Strategic Technology Development Program funded by the Ministry of Knowledge Economy (MKE, Korea)(10039239).

Author details
[1]School of Electronic Engineering, Soongsil University, 369 Sangdo-ro, Dongjak-gu, Seoul 56-743, Korea. [2]School of Computer Science and Engineering, Chung-Ang University, 84 Heukseokro, Dongjak-gu, Seoul 156-756, Korea.

Received: 16 November 2012 Accepted: 11 October 2013
Published: 9 November 2013

References

1. I Stojmenovic, *Handbook of Sensor Networks* (Wiley-Interscience, Hoboken, NJ, 2005)
2. P Santi, Topology control in wireless ad hoc and sensor networks. ACM Comput. Surv. **37**(2), 164–194 (2005)
3. P Santi, The critical transmitting range for connectivity in mobile ad hoc networks. IEEE Trans. Mobile Comput. **4**(3), 310–317 (2005)
4. F Xue, P Kumar, The number of neighbors needed for connectivity of wireless networks. Wireless Netw. **10**(2), 169–181 (2004)
5. N Li, J Hou, L Sha, Design and analysis of an MST-based topology control algorithm, in *Infocom* (IEEE, San Francisco, CA, 30 March–03 April 2003)
6. M Bahramgiri, M Hajiaghayi, V Mirrokni, Fault-tolerant and 3-dimensional distributed topology control algorithms in wireless multi-hop networks, in *ICCCN* (IEEE, Miami FL, 14–16 October 2002)
7. M Hajiaghayi, N Immorlica, VS Mirrokni, Power optimization in fault-tolerant topology control algorithms for wireless multi-hop networks, in *MobiCom* (ACM, San Diego, CA, 14–19 September 2003)
8. X Jia, D Kim, D Wan, C Yi, Power assignment for k-connectivity in wireless ad hoc networks, in *Infocom* (IEEE, Miami, FL, 13–17 March 2005)
9. G Calinescu, PJ Wan, Range assignment for high connectivity in wireless ad hoc networks, in *AdHocNOW* (Springer, Montreal, 08–10 October 2003)
10. N Li, JC Hou, FLSS: a fault-tolerant topology control algorithm for wireless networks, in *MobiCom* (ACM, Philadelphia, PA, 26 September–01 October 2004)

11. R Ramanathan, R Rosales-Hain, Topology control of multihop wireless networks using transmit power adjustment, in *Infocom* (IEEE, Tel Aviv, 26–30 March 2000)
12. M Cardei, S Yang, J Wu, Fault-tolerant topology control for heterogeneous wireless sensor networks, in *MASS* (IEEE, Pisa, 8–11 October 2007)
13. P Corke, P Valencia, P Sikka, T Wark, L Overs, Long-duration solar-powered wireless sensor networks, in *EmNets* (ACM, Cork, 25–26 June 2007)
14. M Minami, T Morito, H Morikawa, T Aoyama, Solar Biscuit: A battery-less wireless sensor network system for environmental monitoring applications, in *INSS* (IEEE, San Diego, CA, 27–28 June 2005)
15. F Simjee, PH Chou, Everlast: long life, supercapacitor operated wireless sensor node, in *ISLPED* (ACM, Tegernsee, 04–06 October 2006)
16. J Taneja, J Jeong, D Culler, Design, modeling and capacity planning for micro-solar power sensor networks, in *IPSN* (ACM, St Louis, 22–24 April 2008)
17. D Noh, D Lee, H Shin, QoS-aware geographic routing for solar-powered wireless sensor networks. IEICE Trans Commun. **E90-B**(12), 3373–3382 (2007)
18. D Noh, I Yoon, H Shin, Low-latency geographic routing for asynchronous energy-harvesting wsns. Int. J. Netw. **3**(1), 78–85 (2008)
19. Y Yang, L Wang, DK Noh, HK Le, T Abdelzaher, Solar Store: enhancing data reliability in solar-powered storage-centric sensor networks, in *MobiSys* (ACM, Kraków, 22–25 June 2009)
20. A Kansal, J Hsu, S Zahedi, MB Srivastava, Power management in energy harvesting sensor networks. ACM Trans. Embedded Comput Syst. **6**(4), 1–38 (2007)
21. M Penrose, On k-connectivity for a geometric random graph. Random Struct. Algorithms. **15**(2), 145–164 (1999)
22. M Marina, S Das, Routing performance in the presence of unidirectional links in multiphop wireless networks, in *Mobihoc* (ACM, Lausanne, 09–11 June 2002)
23. T Lim, G Mohan, Energy aware geographical routing and topology control to improve network lifetime in wireless sensor networks, in *Broadnets* (IEEE, Boston, MA, 03–07 October 2005)
24. L Wang, Y Yang, DK Noh, HK Le, T Abdelzaher, AdaptSens: an adaptive data collection and storage service for solar-powered sensor networks, in *RTSS* (IEEE, Washington, DC, 01–04 December 2009)
25. T Melodia, D Pompili, I Akyildiz, Optimal local topology knowledge for energy efficient geographical routing in sensor networks, in *Infocom* (IEEE, Hong Kong, 07–11 March 2004)

doi:10.1186/1687-1499-2013-258
Cite this article as: Noh and Hur: **Energy-aware topology control for reliable data delivery in solar-powered WSNs.** *EURASIP Journal on Wireless Communications and Networking* 2013 **2013**:258.